

Learn How to Calculate Future Dates Excluding Weekends and Holidays in Excel

Authored by
Mohammed looti

November 10, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learn How to Calculate Future Dates Excluding Weekends and Holidays in Excel*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15493>

When managing project timelines or scheduling crucial deadlines in [Excel](#), it is frequently necessary to calculate a future date based only on [working days](#), effectively skipping weekends and public holidays. Standard date addition methods fail to account for these non-working periods. Fortunately, you can utilize the powerful [WORKDAY function](#) to add a specific number of working days to a particular starting date. This function is essential for accurate business scheduling.

The **WORKDAY function** operates by calculating a date that is a specified number of workdays away from the starting date, automatically excluding Saturdays and Sundays. Understanding its syntax is the first step toward leveraging this tool effectively.

Deconstructing the WORKDAY Function Syntax

This function uses a simple yet precise syntax structure that requires two mandatory arguments and one optional argument:

WORKDAY(start_date, days,)

Each argument plays a critical role in defining the calculation parameters:

start_date: This is the initial date from which the calculation begins. It must be a valid [Excel date](#), usually input as a serial number or a cell reference containing a date value. The calculation starts immediately after this date.

days: This specifies the exact number of **working days** you wish to add to the starting date. This value can be positive (to calculate a future date) or negative (to calculate a past date).

holidays: This is an optional argument that allows you to specify a list or range of dates that should also be excluded from the working calendar, in addition to the standard [weekends](#). This is typically used for public holidays or company-specific shutdowns.

By default, the **WORKDAY function** is programmed to skip Saturday and Sunday when counting the specified number of **days**. If the **start_date** itself falls on a weekend or a listed holiday, the function immediately moves to the next working day before beginning the count. The power of this function lies in its ability to handle complex scheduling requirements simply by providing the necessary parameters, particularly the optional list of holidays.

Case Study 1: Determining a Future Date Without Specified Holidays

Consider a scenario where a project beginning on **12/20/2023** has a delivery requirement exactly **10** working days later. We need to determine the specific calendar date of completion, excluding only the standard weekend days (Saturdays and Sundays).

To execute this calculation, we can organize the data in an Excel sheet. We will place the starting date (12/20/2023) in cell **A2** and the required number of working days (10) in cell **B2**. We then

input the basic **WORKDAY** formula into cell **C2** to calculate the resulting date:

=WORKDAY(A2, B2)

This formula is the simplest implementation of the function, relying entirely on the default exclusion of weekends. Since we omitted the optional third argument, **WORKDAY** assumes no additional days off are required for the count.

The following illustration demonstrates the practical application of this function within the spreadsheet environment:

C2						⌵	⋮	✕	✓	<i>fx</i>	=WORKDAY(A2, B2)
	A	B	C	D	E						
1	Starting Date	Working Days	Ending Date								
2	12/20/2023	10	1/3/2024								
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											

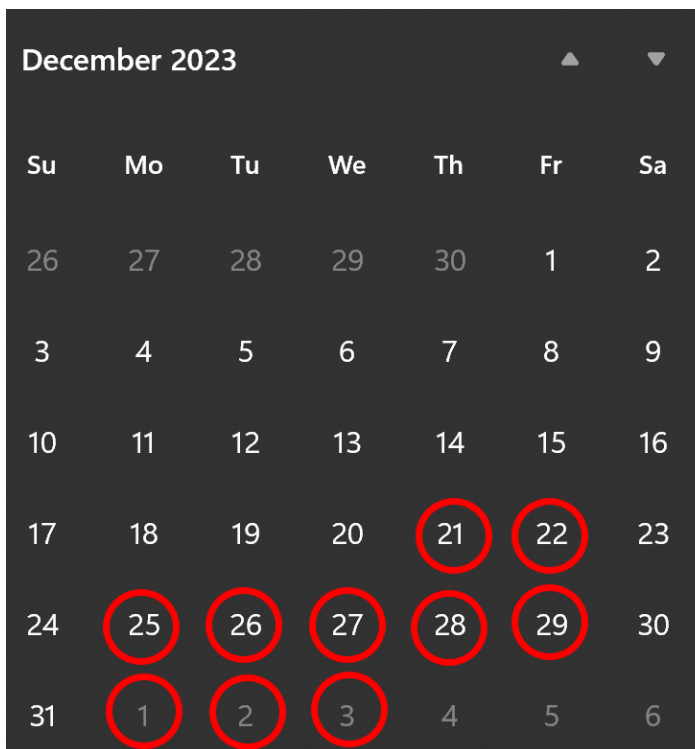
Upon execution, the formula returns the date **1/3/2024**. This date precisely represents the calendar day that falls 10 working days after the starting date of 12/20/2023, after skipping all intervening Saturdays and Sundays. This result confirms that the **WORKDAY** function successfully added 10 week days to the initial date without needing any further intervention regarding holidays.

Verifying the Calculation: How WORKDAY Handles Weekends

It is crucial to understand how the function skips non-working days. When analyzing the calendar between 12/20/2023 and 1/3/2024, the calculation must traverse the end of December and the start of January, a period that includes several weekends. For instance, the weekend of 12/23/2023 and 12/24/2023, and the weekend of 12/30/2023 and 12/31/2023, are automatically excluded from the 10-day count.

We can manually verify this outcome by referencing a calendar and counting forward exactly 10 weekdays from the 12/20/2023 start date. The function accurately performs this tedious counting process instantly.

The visual confirmation below highlights which days were counted as the 10 working days, demonstrating the automatic exclusion of the weekend days:



December 2023						
Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

This verification confirms the default behavior: without the optional **holidays** argument, the **WORKDAY function** only skips the standard Saturday and Sunday work cycle, resulting in the calculated end date of January 3rd.

Case Study 2: Accounting for Non-Standard Days Off Using the Holidays Argument

In many business environments, calculations must account for official public holidays that do not fall on a weekend. Let's revisit the previous example: we still need to add **10** working days to the start date **12/20/2023**, but now we must explicitly exclude two specific holidays: December 25, 2023 (Christmas Day) and January 1, 2024 (New Year's Day).

To accommodate these statutory days off, we must utilize the optional third argument of the **WORKDAY function**. First, we list these holidays in a separate range on the spreadsheet (for example, cells A6 through A7). We then reference this range directly within our formula.

We enter the following enhanced formula into cell **C2**, which incorporates the range of dates to be skipped:

=WORKDAY(A2, B2, A6:A7)

This configuration instructs the function to count 10 working days while skipping Saturdays, Sundays, and any dates found within the designated holiday range (A6:A7). The setup is illustrated in the screenshot below, showing the necessary inputs for the start date, days to add, and the holiday list:

C2						
=WORKDAY(A2, B2, A6:A7)						
	A	B	C	D	E	
1	Starting Date	Working Days	Ending Date			
2	12/20/2023	10	1/5/2024			
3						
4						
5	Holidays					
6	12/25/2023					
7	1/1/2024					
8						
9						
10						
11						
12						
13						

As a result of including the holidays, the formula now returns **1/5/2024**. This is two days later than the previous result (1/3/2024), precisely because the two specified holidays (12/25/2023 and 1/1/2024) were non-weekend working days that had to be skipped over during the 10-day count.

Advanced Considerations and Common WORKDAY Function Pitfalls

The inclusion of the holiday argument demonstrates the robust nature of the **WORKDAY** function for professional scheduling. When the function iterates through the days, it checks against both the fixed weekend rule and the dynamic holiday list simultaneously. If a holiday falls on a Saturday or Sunday, it is effectively skipped only once, as the weekend rule handles the exclusion.

A verification of the calculation, using a calendar that highlights the explicitly skipped holidays alongside the standard weekends, confirms the new end date:

December 2023						
Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

As is clearly shown, the **WORKDAY** function successfully added 10 week days to 12/20/2023, accurately skipping over 12/25/2023 and 1/1/2024, alongside the standard weekends, resulting in the final value of **1/5/2024**. It is important to note that if your required weekend definition differs (e.g., Friday/Saturday instead of Saturday/Sunday), you should use the related function **WORKDAY.INTL**, which allows for customization of the weekend days.

Further Exploration of Date and Time Functions

The **WORKDAY** function is just one component of Excel's powerful suite of date and time manipulation tools. Mastering this function is critical for anyone involved in project management, finance modeling, or resource planning where time-sensitive calculations are required.

For users seeking more comprehensive detail on parameters, error handling, or alternative date calculation methods (such as calculating the number of working days between two dates using **NETWORKDAYS**), the official documentation serves as the most reliable resource.

The following tutorials explain how to perform other common tasks in Excel: